PLACE: Physical Layer Cardinality Estimation for Large-Scale RFID Systems

Yuxiao Hou†, Jiajue Ou†, Yuanqing Zheng‡, Mo Li†
†School of Computer Engineering, Nanyang Technological University, Singapore
‡Department of Computing, The Hong Kong Polytechnic University
{s120003, jou1, limo}@ntu.edu.sg, csyqzheng@comp.polyu.edu.hk

***Abstract*—Estimating the number of RFID tags is a fundamental operation in RFID systems and has recently attracted wide attentions. Despite the subtleties in their designs, previous methods estimate the tag cardinality from the slot measurements, which distinguish idle and busy slots and based on that derive the cardinality following some probability models. In order to fundamentally improve the counting efficiency, in this paper we introduce PLACE, a physical layer based cardinality estimator. We show that it is possible to extract more information and infer integer states from the same slots in RFID communications. We propose a joint estimator that optimally combines multiple sub-estimators, each of which independently counts the number of tags with different inferred PHY states. Extensive experiments based on the GNURadio/USRP platform and the large-scale simulations demonstrate that PLACE achieves approximately 3∼4× performance improvement over state-of-the-art cardinality estimation approaches.**

I. Introduction

Radio Frequency IDentification (RFID) technologies [19] have been developing rapidly in recent years. Due to the low cost and small form factor of RFID tags, RFID technology is widely used to label a large number of items and support inventory management [24], item tracking [14], access control [3], etc.

Counting the number of tags is a fundamental operation. Knowing the tag cardinaltiy can facilitate many primary operations in RFID systems such as tag identification [25] and tag searching [27]. An estimation with guaranteed accuracy normally suffices for the practical purposes. As such, many probabilistic counting methods [4, 8, 10, 11, 15, 16, 26, 29] trade the estimation accuracy for the execution time. Previous works typically measure the states of f communication slots, where each tag responds in one random slot. The slot state can be binary if we distinguish busy and idle slots, or it can be ternary if we further differentiate singleton and collision slots. Thus, the tag responses in f slots can be represented with a $f \times 1$ binary or ternary sequence, with zeros representing idle slots. Intuitively, when a larger number of tags participate, we expect more tag responses and consequently fewer idle slots in the response sequence. Despite the subtleties in design details, previous methods estimate the tag cardinality by examining the state of each slot in the response frame and following a probability model to derive the cardinality.

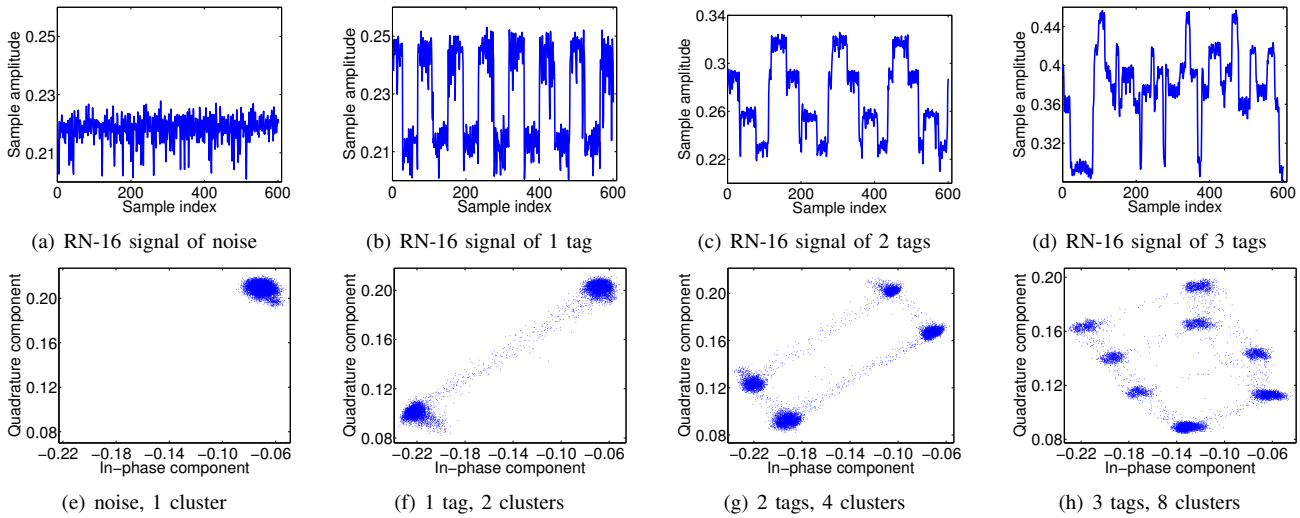
For instance, one previous work EFNEB [8] uses the first busy slot to estimate, while ZOE [29] computes the ratio of zero entries over f slots and derives the tag cardinality. The most recent work [4] advocates the importance of two-phase estimation, and approaches theoretical optimal performance with the binary responses. As only 1 bit or slightly more information is extracted from each slot, previous methods need substantial number of slot measurements to guarantee an estimation accuracy.

In this paper, we present PLACE, a Physical LAyer Cardinality Estimation scheme which extracts more information from each tag response slot, thereby achieving higher estimation efficiency. Unlike previous methods which only distinguish binary or ternary states in each slot, we show that it is possible to detect the number of concurrent tag responses and thus infer integer states from the same slot at RFID physical layer.

To illustrate the possibility of detecting integer slot states, Fig. 1 plots real received signals when 0, 1, 2, and 3 tags respond in the same slot (where Fig. 1(a)-(d) depict time domain signal strengths and Fig. 1(e)-(h) depict I-Q plane signal constellation maps). These traces are collected with our measurement testbed including the GNURadio/USRP2 platform and WISP tags (testbed settings detailed in Section II). While straightforward measurement of signal strength levels in time domain can only tell busy or idle state of the slot, we observe from Fig. 1(e)-(h) that if k tags reply at the same time, 2^k symbol clusters are clearly formed in the corresponding constellation map. This is because each tag takes one of the two states by either reflecting or absorbing radio waves from the RFID reader. Such observation inspires us to detect the exact number of concurrent tag responses in each slot. Ideally, we can infer the number of responding tags from the number of clusters formed, and thereby extend the binary or ternary sequence to an integer sequence.

Although simple in concept, the implementation of physical layer estimation entails many practical challenges. (1) Accurate and efficient estimation of the symbol clusters is non-trivial. In particular, the symbol clustering and counting operation has to be accommodated into the time frame of each RFID slot. In this paper, we design a slot state detection algorithm that divides the I-Q plane into grids and derives the number of symbol clusters based on the symbol densities of grids. The proposed SSDA algorithm takes only millisecond time in comparison with general clustering algorithms that may take hundreds of seconds. (2) Novel cardinality estimator needs to be designed to make the best use of sequences of integer-

978-1-4799-8381-0/15/$31.00 ©2015 IEEE



(a) RN-16 signal of noise (b) RN-16 signal of 1 tag (c) RN-16 signal of 2 tags (d) RN-16 signal of 3 tags

(e) noise, 1 cluster (f) 1 tag, 2 clusters (g) 2 tags, 4 clusters (h) 3 tags, 8 clusters

Fig. 1. Tag response signals and their corresponding constellation maps.

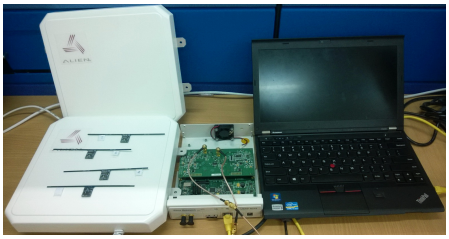


Fig. 2. Our testbed with a GNURadio/USRP2 platform and 4 WISP tags.

state transmission slots instead of previous busy/idle slots. The proposed PLACE scheme combines multiple estimations obtained from each integer slot state and uses an optimal joint estimator such that the overall variance is minimized. (3) Due to noises in practical RFID transmissions, the cluster estimation output inherently contains errors. We run full experiments to understand such errors and analyze the impact on final cardinality estimation accuracy. Our analysis and experiments show that the developed probabilistic estimators in PLACE tolerate the error level from practical measurements.

PLACE is comprehensively evaluated on our testbed built with the GNURadio/USRP platform and WISP tags. We perform large-scale simulations to compare with state-of-the-art cardinality estimation schemes. Experiment and simulation results demonstrate that PLACE achieves approximately 3$\sim$4$\times$ performance improvement.

II. Background

A. Problem description

Following previous works [4, 8, 10, 11, 15, 16, 24–27, 29], we consider a large-scale RFID system consisting of a number of RFID tags covered by one RFID reader. The RFID systems may use lightweight passive RFID tags or powerful active tags.

We exclusively study the RFID communications working at the 900MHz UHF band. Current commodity RFID systems adopt the frame-slotted Aloha model, where a frame is divided into a number of slots. Each of RFID tags randomly chooses one slot in the frame to reply. As a result, one slot might be idle, if no tag responds in the slot; or busy, if at least one tag responds. Instead of replying with a 96-bit tag ID [1], which is used in identification-based counting method [25], each tag only needs to reply with an RN16 sequence in probabilistic cardinality estimation approaches.

Suppose the actual tag cardinality is t, and our estimation is $\hat{t}$. A user-specified accuracy requirement (ϵ,δ) can be specified as follows:

$$Pr\{|\hat{t} - t| \leq \epsilon t\} \geq 1 - \delta. \quad (1)$$

For instance, if the actual number of tags is 1000 and a user specifies the requirement as (5%,1%), then the estimation result is expected to be within the interval [950,1050] with a probability $\geq$ 99%. An ideal estimation approach is expected to meet the estimation accuracy with the minimum execution time.

Many works try to improve the cardinality estimation efficiency [4, 8, 10, 11, 15, 16, 26, 29]. Despite the differences in design details, these works estimate the tag cardinality by measuring the slot states and differentiating idle and busy slots, where each tag randomly selects a slot and sends a short message. For instance, EFNEB [8] infers the tag cardinality from the position of the first busy slot. ZOE [29] computes the ratio of idle and busy slots and thereby derives the tag cardinality. ART [16] measures the average run of busy slots to estimate the tag cardinality. Above approaches [8, 16, 29] adopt the two-phase estimation, where in the first phase the system parameters are optimized to ensure high estimation efficiency in the second phase. One most recent work [4] gives an in-depth analysis and explicitly emphasizes the importance of the two-phase design. To the best of our knowledge, all existing works do not leverage the RFID physical layer information. They only extract binary information from each short slot in the frame.

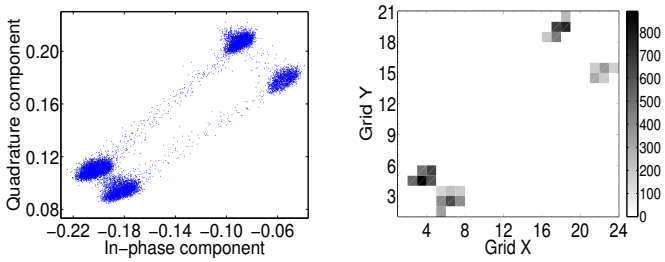
B. Initial observation from our software defined testbed

To explore the possibility of cardinality estimation with PHY layer information of RFID transmissions, we set up a testbed with the GNURadio/USRP software defined radio and the WISP tags as depicted in Fig. 1. We use one USRP RFX900 daughterboard working at the 900MHz UHF band to down-convert the radio signals to the base band. After the down-conversion, the physical layer symbols are transferred to a laptop via a gigabit ethernet link for digital processing. The physical layer sampling rate of the software defined RFID reader is set to 4 million samples per second (MS/s). Thus, the software reader samples 4000 physical symbols every 1ms. At the physical layer, the USRP reader can retrieve the in-phase and quadrature components of each received symbol, which corresponds to a sample at the I-Q plane.

For each RN16 transmission, a commodity tag needs to send a preamble prior to the RN16 payload. The transmission time of an RN16 varies from 0.02ms to 8ms, depending on the backscatter link frequency (BLF) as well as the coding scheme (e.g., FM0, Miller-4) [1]. In our software testbed, the WISP tags are programmed to encode the RN16 messages with Miller-4 and backscatter at 64kbps, which takes around 2ms.

We collect more than 200 physical layer traces when different number of tags concurrently transmit RN16 messages. In Fig. 1, we present 4 instances of received RN16 slots with different number of responding tags. In Fig. 1(a)-(d), for illustration purposes, we intercept the first 600 samples (corresponding to preambles of RN16) of the approximately 8000 samples of each trace. Fig. 1(a)-(d) plot the magnitudes of received symbols. Fig. 1(a) measures the background noise when no tag transmits in the slot. When one tag backscatters its RN16 by reflecting or absorbing radio signals, as shown in Fig. 1(b), the received signal strength at the reader may vary depending on the message content. Current commodity readers set an empirical magnitude threshold to decode the backscattered message. When 2 tags transmit simultaneously, we observe the tag collisions as in Fig. 1(c). In such a case, commodity readers cannot reliably decode the tag collisions, since the threshold based method no longer works. We cannot differentiate the number of colliding tags when more than 2 tags transmit together as in Fig. 1(c) and Fig. 1(d), by solely examining the magnitude of received signals.

When we examine the physical symbols in the I-Q plane as depicted in Fig. 1(e)-(h), however, we see that the symbols exhibit distinct clustering patterns, depending on the number of colliding tags. Fig. 1(e) plots the physical layer symbols that are measured when no tag transmits. If there is no noise, all physical layer symbols overlap at one point in the I-Q plane. In practice, due to background noise, the symbols will be dispersed. As the noise generally follows the Gaussian distribution, the symbols are still clustered and concentrated around one centroid point as in Fig. 1(e). When 1 tag backscatters alone, 2 clusters emerge in the I-Q plane as in Fig. 1(f). Each cluster represents one possible transmission state, i.e., idle or backscattering. We notice that a few samples locate in a narrow band between the two clusters, because the tag takes very short time to transit between the two transmission states. When 2 tags transmit simultaneously, we find that the I-Q plane contains 4 clusters as in Fig. 1(g). This is because we have 4 possible transmission states when 2 tags transmit simultaneously. In Fig 1(h), we see that the number of clusters doubles as one more tag joins in the transmission. Comparing Fig. 1(g)-(h) with Fig. 1(c)-(d), we see that the clustering pattern of physical layer symbols in the I-Q plane contains substantially richer information that allows us to derive the number of colliding tags in each slot.



(a) Physical layer symbols. (b) Filtered grid density matrix.

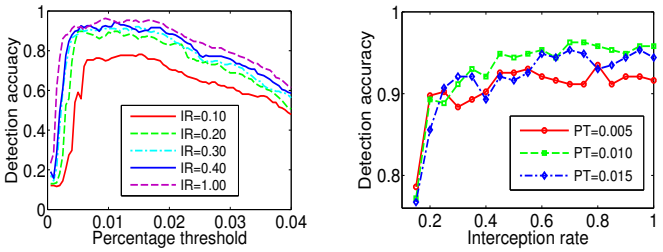
Fig. 3. An illustrative example of Slot State Detection Algorithm (SSDA).

III. Slot State Detection Algorithm

From the initial experiments, we see that it is possible to infer the number of colliding tags by clustering the physical symbols in the I-Q plane. Traditional clustering algorithms, however, are inadequate to serve our purpose for at least two reasons. First, many clustering algorithms (e.g., k-means [13]) require a priori knowledge of the number of clusters. Obviously, such algorithms cannot be directly used since the number of clusters is exactly the unknown that we need to derive. Second, although some clustering algorithms (e.g., DBSCAN [6]) do not require the priori knowledge of the cluster number, they typically incur high computational overhead. In particular, DBSCAN incurs a computation overhead of $O(l^2)$, where l denotes the number of input samples. In our system, we need to cluster thousands of symbols in at most 2ms, i.e., the RN16 slot length.

In this section, we propose a slot state detection algorithm (SSDA) to efficiently process the physical layer symbols and accurately measure the number of colliding tags in the slot. The proposed SSDA method only incurs a computation overhead of $O(l)$. Intuitively, SSDA leverages the fact that samples in one cluster follow a 2-D Gaussian distribution due to channel noise, and consequently we expect a peak grid for each cluster. Thus, we measure the density of samples in each small grid and count the number of clusters by searching the local maximum in the I-Q plane. The input to SSDA is the physical symbols sampled in one slot. The output of SSDA is the number of responding tags in the slot. The whole process of SSDA contains the following three steps:

Step 1: Calculate the sample density in the I-Q plane. We first find the min and max values of both in-phase and quadrature components over all the physical symbols, represented



(a) Detection accuracy with different percentage thresholds (PT).

(b) Detection accuracy with different interception rates (IR).

Fig. 4. Detection accuracies of SSDA with different system parameters.

with complex numbers. We then divide the rectangular area into small grids, whose size is set to 0.01×0.01. We calculate the symbol density of each grid by counting the number of symbols within the corresponding grid.

Step 2: Filter out the noise grids. We set an empirical threshold to differentiate all grids into signal grids and noise grids. If the density of a grid is above the threshold, the grid is considered as a signal grid; otherwise, we filter out the noise grid. Fig. 3 depicts how the raw samples of physical symbols in Fig. 3(a) are filtered to obtain the density matrix in Fig. 3(b). A constant threshold cannot work well since the grid densities may vary, depending on the number of samples as well as the number of clusters that are formed in the I-Q plane. Thus, we propose to use a percentage threshold PT as follows. Suppose there are l data samples in one slot. We set the grid density threshold to be $PT \times l$ for a percentage threshold of PT.

Step 3: Calculate the number of responding tags. We count the number of clusters C by counting the number of the local maximums of sample density in the I-Q plane. In principle, if k tags collide together in a slot, we should observe $C = 2^k$ clusters. Due to noise, we have $C \leq 2^k$ in practice. Thus, we compute the number of responding tags k as $\lceil \log_2 C \rceil$.

In the following, we study the impact of two key system parameters in SSDA – the percentage threshold, and the number of physical samples collected in one slot. The empirical percentage threshold of SSDA influences detection accuracy in practice: a high threshold may miss the cluster peaks, while a low threshold cannot adequately filter out noise grids.

The number of samples collected in one slot is determined by the slot duration and the sampling rate. As the C1G2 standard supports different combinations of backscatter link frequencies and modulation schemes, we can reduce the slot duration with higher link frequencies and coding rates, so as to reduce the transmission time and the computation overhead involved in SSDA.

We carry out trace-driven evaluations to study the influence of the two system parameters. We sample the physical symbols at 4MS/s on our testbed and the slot duration is 2ms. We intercept varied portions of the symbols as input to SSDA. We use an interception rate IR to represent the intercepted portion, e.g., an interception rate IR of 10% means only the first 10% of samples are processed by SSDA. We program the WISP tags and let different number of tags concurrently send RN16 messages in each slot. We record the number of responding tags as the ground truth and measure the detection accuracy. The accuracy is defined as the ratio of correctly detected slots to the number of tested slots.

Fig. 4(a) plots the detection accuracy with the varied percentage threshold ranging from 1% to 4%. We measure the detection accuracy with 5 different interception rates. In the figure, we find that a small percentage threshold (e.g., $<$0.5%) leads to low detection accuracies, because noise grids cannot be filtered out. With the same interception rate, a percentage threshold within [0.5%,1.5%] consistently achieves high detection accuracies. More importantly, we find that the detection accuracy is less sensitive to the change of the percentage threshold within [0.5%,1.5%]. Once the percentage threshold exceeds 2.0%, the detection accuracy decreases as the threshold increases. This is because with a higher threshold, some peak grids with relatively lower grid densities would be accidentally filtered out. Thus, we set the percentage threshold PT to 1%.

We study the impact of interception rates on the detection accuracy. Fig. 4(b) plots the detection accuracy with different interception rates. In the figure, we see that with PT=1% and IR=100%, SSDA can achieve the detection accuracy of above 95%. Moreover, the detection accuracy remains above 95% as long as the interception rate is larger than 50%. In other words, SSDA only needs half of the physical symbols sampled within one slot, to accurately count the number of responding tags. As long as the interception rate is higher than 30%, our detection algorithm can achieve 90% detection accuracy. The experiment results imply that we can potentially reduce the slot duration to further reduce the transmission time.

IV. Estimation Algorithm

As we can differentiate multiple slot states with SSDA, we can devise several estimators for different slot states. For instance, we can estimate the tag cardinality with the fraction of singleton slots, double-tag-collision slots, triple-tag-collision slots, etc. Finally, we design an optimal joint estimator by combining estimations from these subestimators so that the overall variance is minimized.

A. Estimation protocol

In each slot, each of t tags generates a random integer r using a uniform hash function. We denote the index of the right-most zero in the binary representation of r as R. As in the previous schemes [15, 29], a tag will respond if $R \geq \theta$, where θ is a parameter specified by the reader. Therefore, the probability p that a tag will respond in a slot is as follows

$$p = 2^{-\theta}. \tag{2}$$

Suppose X_k $(k = 0, 1, 2, ...)$ is defined as an indicator of k tag responses in a slot, i.e., $X_k = 1$, if k tags are in the slot; $X_k = 0$, otherwise.

For each k, X_k follows the Bernoulli distribution, and the probability of observing k responses in a slot is

$$Pr\{X_k = 1\} = \binom{t}{k} p^k (1-p)^{t-k} \approx \frac{\lambda^k}{k!} e^{-\lambda}, \tag{3}$$

where $\lambda = pt$ is the load factor.

Thus, the expectation $E[X_k]$ and variance $\sigma^2_{X_k}$ are

$$E[X_k] = \frac{\lambda^k}{k!}e^{-\lambda}\ ,\ \sigma^2_{X_k} = \frac{\lambda^k}{k!}e^{-\lambda}(1-\frac{\lambda^k}{k!}e^{-\lambda}). \quad (4)$$

We define $\bar{X_k} = \frac{1}{m}\sum_{i=1}^{m} X_{k,i}$ as the arithmetic average of m observations. Then, the expectation and variance of $\bar{X_k}$, denoted as $E[\bar{X_k}]$ and $\sigma^2_{\bar{X_k}}$, are as follows

$$E[\bar{X_k}] = E[X_k] = \frac{\lambda^k}{k!}e^{-\lambda}\ ,\ \sigma^2_{\bar{X_k}} = \frac{1}{m}\sigma^2_{X_k} \quad (5)$$

Since different k values will produce different estimations of t with different variances, we give the following theorem which provides the optimal combination of multiple sub-estimators.

Theorem 1: Suppose $\hat{t_0}, \hat{t_1}, ..., \hat{t_k}$ are $k+1$ estimations for t with variances $\sigma_0^2, \sigma_1^2, ..., \sigma_k^2$, respectively. For the weighting scheme $\sum_{i=0}^{k} w_i = 1$ and $0 \le w_i \le 1$, the joint estimator $\hat{t} = \sum_{i=0}^{k} w_i \hat{t_i}$ has a variance of $\sigma^2_{\hat{t}} = \sum_{i=0}^{k} w_i^2 \sigma_i^2$. The optimal weights w_i $(i = 0, 1, ..., k)$ for each sub-estimator that minimizes $\sigma^2_{\hat{t}}$ is

$$w_i^* = \frac{1/\sigma_i^2}{\sum_{j=0}^{k} 1/\sigma_j^2}\ ,\ i = 0, 1, ..., k, \quad (6)$$

and the minimum variance is

$$\sigma^2_{\hat{t},min} = \frac{1}{\sum_{i=0}^{k} 1/\sigma_i^2}. \quad (7)$$

Proof: To minimize $\sigma^2_{\hat{t}}$, we define the following Lagrange multiplier

$$L(w_0, w_1, ..., w_k, \beta) = \sum_{i=0}^{k} w_i^2\sigma_i^2 + \beta(\sum_{i=0}^{k} w_i - 1),$$

where the term $\sum_{i=0}^{k} w_i - 1$ incorporates the weight constraint $\sum_{i=0}^{k} w_i = 1$.

We let the partial derivatives of L over w_i $(i = 0, 1, ..., k)$ and β be 0. Thus, we have

$$\begin{cases} \frac{\partial L}{\partial w_i} = 2w_i\sigma_i^2 + \beta = 0\ ,\ i = 0, 1, ..., k \\ \frac{\partial L}{\partial \beta} = \sum_{i=0}^{k} w_i - 1 = 0. \end{cases}$$

We solve the equations as follows

$$\begin{cases} w_i^* = \frac{1/\sigma_i^2}{\sum_{j=0}^{k} 1/\sigma_j^2}\ ,\ i = 0, 1, ..., k. \\ \beta^* = -\frac{2}{\sum_{i=0}^{k} 1/\sigma_i^2}. \end{cases}$$

Thus, we have the minimum variance of $\hat{t}$ as follows

$$\sigma^2_{\hat{t},min} = \sum_{i=0}^{k} w_i^{*2}\sigma_i^2 = \frac{1}{\sum_{i=0}^{k} 1/\sigma_i^2}.$$

■

B. Computing the number of rounds m

In practice, m estimation rounds have to be repeated to further reduce $\sigma^2_{\hat{t},min}$ and meet the requirement in Eq.(1). In the following, we analyze the minimum value of m.

Let $Y = \frac{\hat{t} - t}{\sigma_{\hat{t},min}}$. Since t is often a large number, according to the law of large number, Y follows the standard normal distribution. Thus, we can derive

$$Pr\{-\frac{\epsilon t}{\sigma_{\hat{t},min}} \le Y \le \frac{\epsilon t}{\sigma_{\hat{t},min}}\} \ge 1 - \delta. \quad (8)$$

Eq.(8) is equivalent to

$$\frac{\epsilon t}{\sigma_{\hat{t},min}} \ge c, \quad (9)$$

where c meets the following condition

$$1 - \delta = erf(\frac{c}{\sqrt{2}}), \quad (10)$$

and $erf()$ represents the Gaussian error function.

Since $\sigma^2_{\hat{t},min}$ is a function of m, we first compute $\sigma^2_{\hat{t},min}$. According to Eq.(7), $\sigma^2_{\hat{t},min}$ is a function of σ_k^2. Because $\hat{t_k}$ is a function of $\bar{X_k}$, the relationship between σ_k^2 and $\sigma^2_{\bar{X_k}}$ can be explicitly expressed.

Suppose $\hat{t_k} = f_k(\bar{X_k})$ is expressed with the Taylor expansion centered on $E[\bar{X_k}]$:

$$f_k(\bar{X_k}) - f_k(E[\bar{X_k}]) \approx f_k'(E[\bar{X_k}])(\bar{X_k} - E[\bar{X_k}]). \quad (11)$$

Since we have $f_k(E[\bar{X_k}]) = E[f_k(\bar{X_k})]$, we derive from Eq.(11) that

$$Var[f_k(\bar{X_k})] \approx \{f_k'(E[\bar{X_k}])\}^2 Var[\bar{X_k}]. \quad (12)$$

We represent Eq.(12) as follows

$$\sigma_k^2 = \alpha_k^2 \sigma^2_{\bar{X_k}}, \quad (13)$$

where

$$\alpha_k = \frac{dt_k}{d\bar{X_k}}\Big|_{E[\bar{X_k}]} = \frac{1}{d\bar{X_k}/dt_k}\Big|_{E[\bar{X_k}]} = \frac{k!te^{\lambda}}{\lambda^k(k-\lambda)}. \quad (14)$$

Combining Eq.(4), Eq.(5), Eq.(7), Eq.(13), and Eq.(14), we obtain the expression of $\sigma^2_{\hat{t},min}$ as follows:

$$\sigma^2_{\hat{t},min} = \frac{t^2}{mg(\lambda)}, \quad (15)$$

where $g(\lambda)$ is

$$g(\lambda) = \sum_{i=0}^{k} \frac{(i-\lambda)^2}{i!e^{\lambda}/\lambda^i - 1} \quad (16)$$

After obtaining $\sigma^2_{\hat{t},min}$, we can derive the number of independent measurements m to meet the accuracy requirement by substituting Eq.(15) into Eq.(9):

$$m \ge \frac{c^2}{\epsilon^2 g(\lambda)}. \quad (17)$$

We see that $g(\lambda)$ depends on k which is the maximum number of detectable colliding tags in one slot. Fig. 5 measures m with different k to meet different (ϵ,δ)-accuracy requirements. We fix ϵ to 0.01 and specify δ to 1%, 10%, and 20%, respectively. In the figure, we find that regardless of the accuracy requirement, PLACE needs to perform fewer rounds

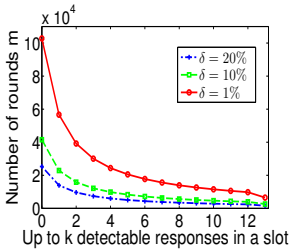


Fig. 5. m against k.

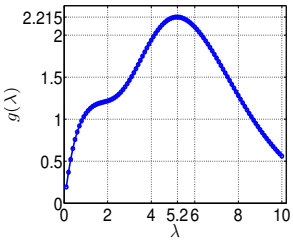


Fig. 6. $g(\lambda)$ against λ.

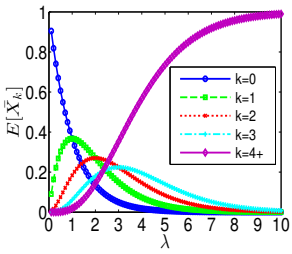


Fig. 7. The expectation of $\bar{X}_k$ over load factor λ.

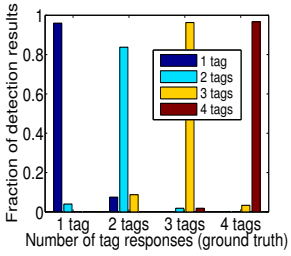


Fig. 8. Detailed accuracy performance of SSDA.

of estimation with the increased k. This is because PLACE is able to augment the joint estimator with more independent sub-estimators. Nevertheless, the marginal gain of detecting more tags in each slot gradually decreases as k increases. As the slot state detection accuracy decreases when more tags collide together, we combine 4 sub-estimators in practice.

We further tune λ to maximize $g(\lambda)$ and minimize m. In order to find the optimal λ^* to maximize $g(\lambda)$ and minimize m, we plot $g(\lambda)$ against λ in Fig. 6. We see that $g(\lambda)$ reaches the maximum value with $\lambda^* \approx 5.2$.

C. Two-phase Counting Algorithm

We adopt the two-phase estimation design [4]. In the first rough estimation phase, we adjust the threshold θ^* so that the load factor $\hat{\lambda}$ approaches 5.2; in the second phase, we repeat m independent estimation rounds with the optimal threshold θ^*. The weights that are necessary to compute the final estimation $\hat{t}$ are derived from $\hat{\lambda}$ obtained in the first phase.

In the first rough estimation phase, the reader issues a θ value and measures the fraction of each slot state, i.e., $\bar{X}_k$ ($k = 0, 1, 2, 3$). We denote the fraction of slots with more than 3 concurrent responses as $\bar{X}_{4+}$. From Eq.(5), we have

$$E[\bar{X}_{4+}] = 1 - \sum_{k=0}^{3} E[\bar{X}_k] = 1 - e^{-\lambda} \sum_{k=0}^{3} \frac{\lambda^k}{k!}. \quad (18)$$

Fig. 7 plots $E[\bar{X}_k]$ ($k = 0, 1, 2, 3, 4+$) with different λ. From this figure we observe that when λ is around 5.2, $E[\bar{X}_k]$ ($k = 0, 1, 2, 3$) can be very small and hence cannot be accurately measured with a small number slots (e.g., 32 slots). In contrast, $E[\bar{X}_{4+}]$ spans a relatively large range and can be a good indicator of λ. In addition, $E[\bar{X}_{4+}]$ monotonically increases with λ, which allows us to quickly converge to the optimal λ using the binary search method.

In particular, in each query round, we measure $E[\bar{X}_{4+}]$ and compute $\hat{\lambda}$ according to Eq.(18). If $\hat{\lambda}$ is smaller than 3, indicating a large θ value, we decrease θ in the next query round; if $\hat{\lambda}$ is larger than 7, indicating a small θ value, we increase θ in the next query round; once $\hat{\lambda}$ is in the range $[3, 7]$, we terminate the rough estimation phase and set θ^* and $\hat{\lambda}$ to the corresponding values in the last query round. Based on the above rules, the reader starts a query round with $\theta = 16$ and adopts a binary search method for θ in the range of $[0, 32]$.

D. Impact of SSDA Errors and Enhancement

We analyze how the SSDA detection errors influence the counting accuracy of PLACE.

We denote q_{ij} as the probability of detecting state i as state j, where $i, j = 1, 2, 3, 4+$. Specifically, if $i = j$, q_{ij} indicates the detection accuracy of state i. As empty slots (state 0) can be accurately differentiated from busy slots by measuring the signal strength, we only consider the detection accuracy of state i, where $i, j = 1, 2, 3, 4+$. We use a detection rate matrix $Q = [q_{ij}]_{4\times4}$ to represent the overall detection performance of SSDA.

We use a vector $\vec{X} = (\bar{X}_1, \bar{X}_2, \bar{X}_3, \bar{X}_{4+})^T$ to represent the actual fraction of each state. As the detection results of SSDA may contain some errors, we represent the measurement results as $\vec{X}^E = (\bar{X}_1^E, \bar{X}_2^E, \bar{X}_3^E, \bar{X}_{4+}^E)^T$. Based on the definition of Q, we have $\vec{X}^E = Q\vec{X}$. Thus, we can obtain $\vec{X}$, which can be used to generate an accurate estimation of t, as follows:

$$\vec{X} = Q^{-1}\vec{X}^E, \quad (19)$$

where Q^{-1} is the inverse matrix of Q.

To estimate Q, we perform SSDA with our traces collected from the software defined testbed, which is described in Section II. We set the percentage threshold to be 1% and the interception rate to be 30%. Fig. 8 plots the state detection accuracy of SSDA. The x-axis of Fig. 8 is the ground truth of each tag response state, and the y-axis represents the detection results. We represent the measurement results with Q as follows:

$$Q = \begin{pmatrix} 0.96 & 0.04 & 0 & 0 \\ 0.08 & 0.84 & 0.09 & 0 \\ 0 & 0.02 & 0.96 & 0.02 \\ 0 & 0 & 0.03 & 0.97 \end{pmatrix}.$$

From the above Q, we find the SSDA method achieves high detection accuracies. For the detection errors, we find that state k is more likely to be mistakenly detected as adjacent states.

In practice, Q can vary due to various factors, e.g., reader transmission power, interference to tag responses, etc. To understand the impact of Q on the overall estimation accuracy of PLACE, we approximate Q with Q_0 as follows:

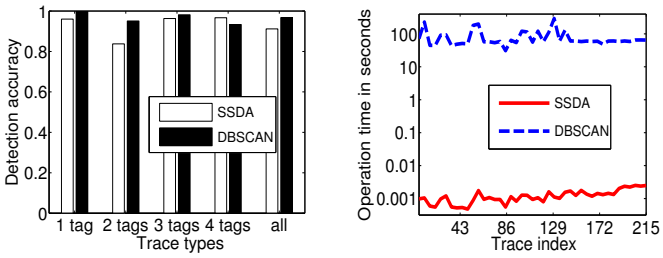
$$Q_0 = \begin{pmatrix} 1-q_0 & q_0 & 0 & 0 \\ q_0 & 1-2q_0 & q_0 & 0 \\ 0 & q_0 & 1-2q_0 & q_0 \\ 0 & 0 & q_0 & 1-q_0 \end{pmatrix},$$

where q_0 can be specified according to empirical measurement results. With Q_0, we can study how different detection performance of SSDA may impact the overall counting accuracy of PLACE. We can recover $\vec{X}$ from $\vec{X}^E$ according to Eq.(19) and use $\vec{X}$ for tag cardinality estimation. We name the enhanced PLACE with the error compensation as EPLACE.

To measure Q_0 in practice, we can first identify k tags and request them to respond together. Then, we can use k as the ground truth and measure each entry in the kth row of Q_0.

V. Evaluation

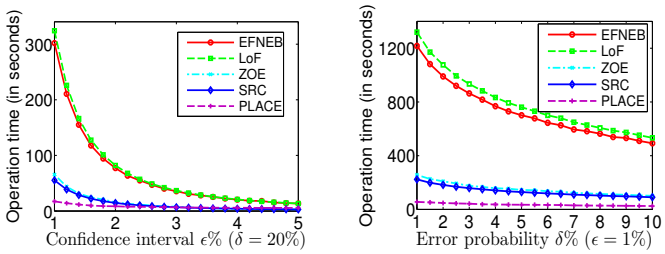
In the following, we first compare SSDA with the benchmark clustering algorithm DBSCAN [6] in terms of the slot state detection accuracy and the execution time. We then compare PLACE with previous cardinality estimation schemes including EFNEB [8], LoF [15], ZOE [29] and SRC [4].



(a) Comparison of detection accuracy.

(b) Comparison of computational overhead

Fig. 9. Performance comparison of DBSCAN and SSDA.



(a) Comparison of operation time with $\delta = 20\%$ and varying ϵ

(b) Comparison of time with $\epsilon = 1\%$ and varying δ

Fig. 10. Comparison of operation time to meet different accuracy requirements among 5 schemes: EFNEB, LoF, ZOE, SRC and PLACE.

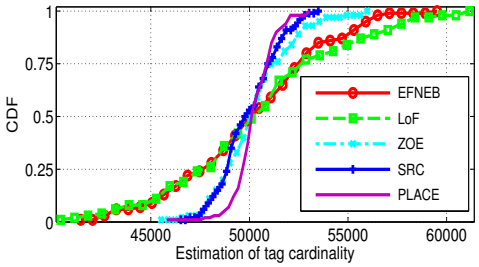


Fig. 11. CDF of estimation results of different schemes with the same amount of execution time.

Finally, we evaluate the impact of SSDA detection errors on the estimation accuracy as well as the compensation for the errors.

A. SSDA Evaluation

Our traces are collected with the GNURadio/USRP testbed and WISP tags as described in Section II. For the experimental purpose, we program the WISP tags and control the number of responding tags in each slot. We record the actual number of responding tags (varying from 1 to 4) as the ground truth in the experiment.

We expect an ideal slot state detection algorithm to efficiently process the samples and accurately count the number of responding tags. We compare the detection accuracy and the execution time of the proposed SSDA with the benchmark scheme DBSCAN. We set the interception rate IR to 30% and the percentage threshold PT to 1% for SSDA.

In DBSCAN, a circular region centered on a point p with radius ε is called ε-neighborhood of p. If at least T points fall into ε-Neighborhood of p, p is called a core point. Otherwise if q falls into the ε-neighborhood of another core point, we call q border point. A noise point is a point that is neither a core point nor a border point. If p is a core point and q is in ε-neighborhood of p, we say q is *directly density-reachable* from p. If q' is directly density-reachable from q, and q is directly density-reachable from p, we say q' is *indirectly density-reachable* from p via q, i.e, $p \rightarrow q \rightarrow q'$. DBSCAN groups all density-reachable points into one cluster. In the experiment, we specify the optimal parameters (ε=0.01, T=0.01$\times l$), which maximize the detection accuracy of DBSCAN. Although DBSCAN can be used to count the number of responding tags in each slot, it incurs a computation overhead of $O(l^2)$, where l denotes the number of input data samples.

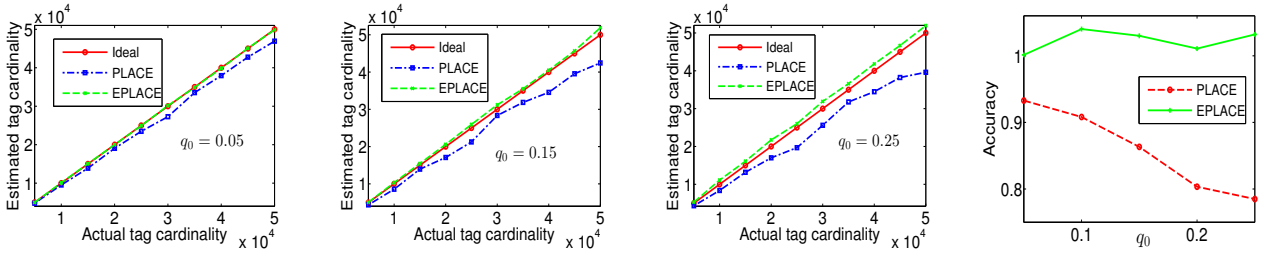
Fig. 9(a) compares the detection accuracy of SSDA and DBSCAN. We present the overall detection accuracy as well as the accuracy for each case with different number of responding tags. Fig. 9(a) shows the following results. First, the overall accuracy of SSDA is comparable with that of DBSCAN. Specifically, the overall accuracies of SSDA and DBSCAN are 91.2% and 96.7%, respectively. Second, in the case when 4 tags respond together, SSDA achieves higher accuracy compared with DBSCAN. This is because when 4 tags respond concurrently, the I-Q plane becomes crowded with 16 clusters. As a result, the inter-cluster distances become smaller and the borders between neighboring clusters become blurred. Thus, the border-based DBSCAN may cluster the neighboring clusters together. In contrast, the centroid-based SSDA overcomes this problem and derives the number of clusters by counting the number of local maximums after filtering out noise. Since the local maximums lie in the center of clusters, the distance between the centers of two neighboring clusters tend to be larger than the distance between their borders.

Fig. 9(b) compares the computational overhead of SSDA and DBSCAN. The physical layer symbols are collected with the USRP reader and the symbols are transferred to a laptop for processing. We execute both algorithms on the laptop and measure the execution time of two algorithms. The laptop is equipped with an Intel qual-core 2.9GHz i7 processor and 15.4GB memory running 64-bit Ubuntu 13.04. In the figure, the x-axis is the trace index and the y-axis is the operation time in seconds, presented in the log scale. We find that SSDA reduces the operation time compared with DBSCAN by orders of magnitude. Specifically, the average operation time of SSDA and DBSCAN is 1.3ms and 84.5s, respectively. SSDA substantially outperforms DBSCAN mainly due to the fact that while DBSCAN incurs $O(l^2)$ computational overhead, SSDA only incurs $O(l)$ overhead. In addition, while DBSCAN has to perform computation-intensive operations such as multiplication and square root calculation to calculate the distance between physical layer symbols, SSDA only needs to perform lightweight operations such as addition and comparison.

B. PLACE Evaluation

We perform extensive simulations to compare PLACE with previous cardinality estimation schemes. As most of these previous schemes do not tolerate noisy channels, we assume no errors in slot state detection in the performance comparison.

We measure the overall execution time as the performance



(a) q_0=0.05 (b) q_0=0.15 (c) q_0=0.25 (d) Accuracy with different q_0.

Fig. 12. Impact of q_0 on estimation accuracy of PLACE.

metric, which counts both communication time and the computation time. The communication time mainly consists of the transmission time of reader's command and tags responses. The computation time is mainly consumed in the execution of SSDA for each slot. We ignore the computation time for benchmark schemes. In practice, as SSDA can be executed in real time, the cluster counting operation (which takes 1.3ms) can be executed in parallel with the signal sampling operation for each RN16 reception (which takes 2ms) at physical layer. Thus, SSDA incurs little extra time overhead.

Fig. 10 compares the overall operation time to meet different estimation accuracy requirements. The actual tag cardinality is 50000. In Fig. 10(a), we fix δ to 20% and vary ϵ, ranging from 1% to 5%. From Fig. 10(a), we find that like benchmark schemes, PLACE takes less time to meet the estimation accuracy requirement of relaxed confidence intervals. Moreover, PLACE takes much less operation time compared with the benchmark schemes to meet the same accuracy requirement. In particular, compared with the recent state-of-the-art work SRC [4], PLACE only takes approximately 1/3 of execution time across different accuracy requirements. In Fig. 10(b), when we fix ϵ to 1% and vary δ from 1% to 10%, we also find that PLACE substantially outperforms benchmark schemes.

We provide each estimation scheme the same amount of execution time to estimate the number of 50000 tags. We repeat the estimation process of each scheme for 100 times. In Fig. 11, we plot the CDF of estimation results for each scheme. From Fig. 11, we find that the estimation results of PLACE are more concentrated on the actual tag cardinality. Moreover, the tail of PLACE is much shorter than those of EFNEB, LoF, ZOE and SRC, indicating smaller estimation variance of PLACE. Specifically, according to the estimation results, provided the same amount of operation time, PLACE has 99 estimation results within the confidence interval [47500,52500], while SRC, which performs best among the benchmarks, has only 91 estimation results within the interval. According to the experiment result, we find that given the same amount of operation time, PLACE can estimate the tag cardinality more precisely and accurately compared with other schemes.

C. The Impact of SSDA Detection Errors on PLACE

To evaluate the impact of SSDA detection errors on the estimation accuracy of PLACE, we measure the estimation accuracy with the ratio of the estimated tag population $\hat{t}$ over the actual population t as in [15, 26, 29]. Ideally, the accuracy should be 1, indicating a perfect estimation result.

We run the basic PLACE and the Enhanced PLACE (EPLACE), which compensates for the errors and adjusts the estimation results. We use q_0 to represent the slot state detection error. We average over 100 runs to obtain each estimation result.

Fig. 12(a)-(c) plot the estimation accuracies of PLACE and EPLACE, with q_0 of 5%, 15%, and 25%, respectively. The y-axis and x-axis represent the estimated number and the actual number of tags, respectively. For illustration purposes, we plot the ideal curve $y = x$. From Fig. 12(a)-(c), we find that without the error compensation, the estimation errors of the basic PLACE increase with both the number of tags and the slot state detection errors. Fortunately, EPLACE is able to compensate for the errors and achieve high accuracy. The experiment results of Fig. 12(a)-(c) demonstrate that EPLACE is able to leverage the knowledge about the detection errors and adjust the estimation results accordingly.

In Fig. 12(d), we vary the error rate from 5% to 25% and measure the corresponding estimation accuracy. We fix the tag cardinality to 50000. We specify the (ϵ=5%,δ=1%)-accuracy requirement, and provide PLACE the corresponding execution time. From Fig. 12(d), we find that as q_0 increases, the estimation accuracy of basic PLACE decreases dramatically. In contrast, the estimation accuracy of EPLACE remains relatively stable and fluctuates around 1. Although EPLACE cannot achieve the ideal estimation accuracy of 1, the estimation results are all within the targeted accuracy interval of [0.95,1.05].

VI. Related Work

Many probabilistic approaches have been proposed to improve the cardinality estimation efficiency [4, 8, 10, 11, 15, 16, 26, 29]. Kodialam *et al.* propose the first probabilistic counting scheme, Unified Probabilistic Estimator [10], which uses the fractions of empty, singleton and collision slots to estimate the tag population. Qian *et al.* propose the Lottery Frame scheme [15] to reduce the frame size and avoid the problem of replicated counting. Han *et al.* present the Enhanced First Non-Zero Based estimator [8], which quickly locates the first busy slot with the binary search. Zheng *et al.* design the Probabilistic Estimating Tree scheme [26], where a binary tree is used to organize the tags and assist the tag probing.

Shahzad *et al.* propose the Average Run based Tag estimator [16], which estimates the tag population with the average run length of non-empty slots. Zheng *et al.* present the Zero-One Estimator [29], where all tags respond in each slot with a certain probability and the fraction of empty slots is used to estimate the tag cardinality. Chen *et al.* [4] emphasize the importance of the two-phase design and study the theoretical limits of RFID counting efficiency. Gong *et al.* [7] efficiently estimate the number of counterfeit tags. Liu *et al.* [12] estimate the number of key tags. Unlike those works that only leverage binary or ternary states extracted from each slot, we propose a cardinality estimation scheme which infers the number of colliding tags in each slot at RFID physical layer and thereby improves the estimation efficiency.

Previous works try to read multiple RFID tags by recovering tag collisions at physical layer [2, 9, 17]. Shen *et al.* [17] propose to use software defined radios to recover collisions of HF RFID cards. Khasgiwale *et al.* [9] decode the RN16 message of UHF RFID tags so as to improve the tag arbitration efficiency. Some works [2, 5] present the theoretical analysis on tag collisions and read a small number of tags in parallel. Nevertheless, such deterministic identification schemes cannot efficiently estimate the tag cardinality for large-scale RFID systems. Inspired by those works, we present a probabilistic estimation scheme which is able to extract and synthesize more information from the RFID physical layer.

Many prior works study the problem of collecting data from RFID devices. Yue *et al.* [21] present a data collection scheme using the Bloom filter. BLINK [23] improves the link layer performance with link quality measurement and rate adaptation for RFID devices. Buzz [18] recovers tag collisions at physical layer and collects data from RFID tags in an efficient and reliable manner. Zanetti *et al.* [22] identify RFID tags using the physical layer fingerprints. P-MTI [28] identifies the missing tags by examining RFID collisions at physical layer. Tagoram [20] tracks mobile tags by leveraging the phase information available at commodity readers. The common rationale of those works and PLACE is that careful cross layer designs of RFID network stack may fundamentally improve the operational efficiency of RFID systems.

VII. Conclusion

Estimating the number of RFID tags is a fundamental operation in RFID systems. In this paper, we introduce a physical layer based cardinality estimator to fundamentally improve the estimation efficiency. We first propose a slot state detection algorithm to accurately count the number of responding tags in each slot. We then devise a joint estimator to combine multiple sub-estimators each of which estimates the tag population with the slot state measurement results. Extensive evaluation results show that PLACE substantially outperforms prior works.

Acknowledgement

We acknowledge the support from Singapore MOE AcRF Tier 1 grant MOE2013-T1-002-005, and NTU Nanyang Assistant Professorship (NAP) grant M4080738.020.

References

[1] EPCglobal C1G2 UHF RFID Protocol at 860MHz-960MHz, http://www.epcglobalinc.org/standards/uhfc1g2, July 2014.

[2] C. Angerer, R. Langwieser, and M. Rupp. RFID reader receivers for physical layer collision recovery. In *IEEE Transactions on Communications*, 58(12):3526-3537, 2010.

[3] G. Avoine and P. Oechslin. A scalable and provably secure hash-based RFID protocol. In *IEEE PerCom Workshops*, 2005.

[4] B. Chen, Z. Zhou, and H. Yu. Understanding RFID counting protocols. In *ACM MobiCom*, 2013.

[5] M. V. B. Delgodo, C. Angerer, J. V. Alonso, M. Rupp. Estimation of the Tag Population with Physical Layer Collision Recovery. In *The Third International EURASIP Workshop on RFID Technology*, 2010.

[6] M. Ester, H. Kriegel, J. Sander, X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM KDD*, 1996.

[7] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, Y. Liu. Informative Counting: Fine-grained Batch Authentication for Large-Scale RFID Systems. In *ACM MobiHoc*, 2013.

[8] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID tags efficiently and anonymously. In *IEEE INFOCOM*, 2010.

[9] R. Khasgiwale, R. Adyanthaya, and D. Engels. Extracting information from tag collisions. In *IEEE International Conference on RFID*, 2009.

[10] M. Kodialam, and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *ACM MobiCom*, 2006.

[11] M. Kodialam, T. Nandagopal, and W. Lau. Anonymous tracking using RFID tags. In *IEEE INFOCOM*, 2007.

[12] X. Liu, K. Li, H. Qi. B. Xiao, and X. Xie. Fast Counting the Key Tags in Anonymous RFID Systems. In *IEEE ICNP*, 2014.

[13] J. MacQueen. Some methods for classification and analysis of multivariate observations. In proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967.

[14] L. Ni, Y. Liu, Y. Lau, and A. Patil. LANDMARC: indoor location sensing using active RFID. *ACM Wireless Networks*, 10(6):701-710, 2004.

[15] C. Qian, H. Ngan, and Y. Liu. Cardinality estimation for large-scale RFID systems. In *IEEE PerCom*, 2008.

[16] M. Shahzad, and A. Liu. Every bit counts - fast and scalable RFID estimation. In *ACM MobiCom*, 2012.

[17] D. Shen, G. Woo, D. Reed, and A. Lippman. Separation of multiple passive RFID signals using software defined radio. In *IEEE RFID*, 2009.

[18] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and Reliable Low-Power Backscatter Networks. In *ACM SIGCOMM*, 2012.

[19] R. Want. An introduction to RFID technology. In *IEEE Pervasive Computing*, 5(1):25-33, 2005.

[20] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. In *ACM MobiCom*, 2014.

[21] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen. A Time-efficient Information Collection Protocol for Large-scale RFID Systems. In *IEEE INFOCOM*, 2012.

[22] D. Zanetti, B. Danev, and S. Čapkun. Physical-layer Identification of UHF RFID Tags. In *ACM MobiCom*, 2010.

[23] P. Zhang, J. Gummeson, and D. Ganesan. BLINK: A High Throughput Link Layer for Backscatter Communication. In *ACM MobiSys*, 2012.

[24] R. Zhang, Y. Liu, Y. Zhang, and J. Sun. Fast identification of the missing tags in a large RFID system. In *IEEE SECON*, 2011.

[25] B. Zhen, M. Kobayashi, and M. Shimizu. Framed ALOHA for multiple RFID objects identification. In *IEICE Transactions on Communications*, E88-B(3), 2005.

[26] Y. Zheng, M. Li, and C. Qian. PET: probabilistic estimating tree for large-scale RFID estimation. In *IEEE ICDCS*, 2011.

[27] Y. Zheng, M. Li. Fast tag searching protocol for large-scale RFID systems. In *IEEE ICNP*, 2011.

[28] Y. Zheng, M. Li. P-MTI: Physical-layer Missing Tag Identification via Compressive Sensing. In *IEEE INFOCOM*, 2013.

[29] Y. Zheng, M. Li. ZOE: fast cardinality estimation for large-scale RFID systems. In *IEEE INFOCOM*, 2013.